We Claim:

1. In a graphics system including a graphics processing pipeline that renders and displays images at least in part in response to primitive vertex data and texture data, a texture processing system for mapping a texture to a surface of a rendered image object, said texture processing system comprising:

a texture coordinate/data processing unit that interleaves processing of logical direct and indirect coordinate data;

a texture data retrieval unit connected to the coordinate/data processing unit, the texture data retrieval unit retrieving texture data; and

a data feedback path from the texture data retrieval unit to the texture coordinate/data processing unit to allow reuse of the texture coordinate/data processing unit in the same rendering pass;

wherein in response to a set of indirect texture coordinates the retrieval unit recirculates retrieved texture data back to the processing unit for deriving modified texture coordinates which are used in mapping a texture to a surface of a rendered image object.

2. The graphics system as set forth on claim 1 wherein the texture coordinate/data processing unit further comprises a set of hardware control logic registers coupled to data lines in the pipeline for receiving data and processing command information used to initiate indirect texture referencing and to control multiplication and addition operations for deriving said modified texture coordinates.

3. In a graphics system having a memory containing texture data, a method of indirect texture referencing comprising the steps of:

(a) using indirect texture coordinates to generate a data triplet;

(b) deriving texture coordinates based at least in part on the generated triplet; and

(c) using at least the derived texture coordinates to map predetermined texture data onto a primitive.

4. A method of indirect texture referencing as in claim 3 wherein the using step (a) includes referencing, in a texture memory, an array that maps the triplet via the indirect texture coordinates.

5. A method of indirect texture referencing as in claim 3 wherein the using step (a) includes fetching predetermined triplet data from a texture map stored in memory.

6. A method of indirect texture referencing as in claim 3 wherein the using step (c) includes referencing, in a texture memory, an array that maps color value data via the derived texture coordinates.

7. A method of indirect texture referencing as in claim 3 wherein the deriving step (b) performs a matrix multiplication operation wherein a first matrix comprising a plurality of predetermined constant and/or variable scalar elements is multiplied by a second matrix comprising a retrieved data triplet.

8. A method of indirect texture referencing as in claim 3 wherein a triplet comprises s, t and u texture coordinate offsets.

9. A method of indirect texture referencing as in claim 3 wherein a triplet comprises three eight-bit binary values.

10. A method of indirect texture referencing as in claim 3 wherein a triplet comprises three five-bit binary values.

11. A method of indirect texture referencing as in claim 3 wherein a triplet comprises three four-bit binary values.

12. A method of indirect texture referencing as in claim 3 wherein a triplet comprises three three-bit binary values.

13. A method of indirect texture referencing as in claim 3 wherein a quadruplet of indirect texture coordinates are processed together to produce a quadruplet of derived texture coordinates for mapping texture data to polygons.

14. A method of indirect texture referencing as in claim 7 wherein said first matrix comprises elements that are a scalar function of one or more predetermined texture coordinates.

15. A method of indirect texture referencing as in claim 7 wherein said first matrix is a 3 X 2 matrix comprising six predetermined scalar elements.

16. A method of indirect texture referencing as in claim 7 wherein said first matrix is arranged having elements as follows:

$$\begin{pmatrix} s/256 & t/256 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined texture coordinates.

17. A method of indirect texture referencing as in claim 7 wherein said first matrix is arranged having elements as follows:

$$\begin{pmatrix} 0 & 0 \\ s/256 & t/256 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined texture coordinates.

18. In a 3D videographics system having a memory containing texture data stored in a texture memory, the texture data accessed via either a set of direct texture coordinates or a set of indirect texture coordinates, a method of indirect texture referencing for mapping a predetermined texture onto a polygon comprising the steps of:

(a) using a set of indirect texture-coordinates to retrieve a data triplet stored in texture memory;

(b) deriving a set of modified texture coordinates based at least in part on the retrieved data triplet; and

(c) using the set of modified texture coordinates to reference texture data stored in texture memory corresponding to the predetermined texture.

19. A method of indirect texture referencing as in claim 18 wherein the deriving step (b) includes performing at least one matrix multiplication operation wherein a first matrix comprising a plurality of predetermined constant and/or variable scalar elements is multiplied by a second matrix comprising a retrieved data triplet.

20. A method of indirect texture referencing as in claim 18 wherein the using step (c) includes referencing, in a texture memory, an array that maps color value data via the derived texture coordinates.

21. A method of indirect texture referencing as in claim 18 wherein a data triplet comprise at least one texture coordinate offset.

22. A method of indirect texture referencing as in claim 18 wherein a data triplet comprises s, t, and u texture coordinate offset values.

23. A method of indirect texture referencing as in claim 18 wherein a data triplet comprises three eight-bit binary values.

24. A method of indirect texture referencing as in claim 18 wherein a data triplet comprises three five-bit binary values.

25. A method of indirect texture referencing as in claim 18 wherein a data triplet comprises three four-bit binary values.

26. A method of indirect texture referencing as in claim 18 wherein a data triplet comprises three three-bit binary values.

27. A method of indirect texture referencing as in claim 19 wherein said first matrix comprises elements that are a scalar function of one or more predetermined direct texture coordinates.

28. A method of indirect texture referencing as in claim 19 wherein said first matrix is a 3 X 2 matrix comprising six predetermined scalar elements.

29. A method of indirect texture referencing as in claim 19 wherein said first matrix is arranged having elements as follows:

$$\begin{pmatrix} s/256 & t/256 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined current direct texture coordinates.

30. A method of indirect texture referencing as in claim 19 wherein said first matrix is arranged having elements as follows:

$$\begin{pmatrix} 0 & 0 \\ s/256 & t/256 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined current direct texture coordinates.

31. In a graphics system including a graphics processing pipeline that renders and displays images at least in part in response to polygon vertex data and texture data stored in a memory, the graphics processing pipeline having a texture subsystem for accessing and retrieving texture. the texture subsystem comprising a texture coordinate/data processing unit having: a) at least one binary data multiplier, at least one binary data accumulator and at least one control register for receiving instruction codes and/or data to control texture coordinate/data processing operations, b) a texture data retrieval unit connected to the coordinate/data processing unit, the texture data retrieval unit retrieving texture data stored in a texture memory, and c) a data feedback path from the texture data retrieval unit to the texture coordinate/data processing unit to recycle retrieved texture data through the texture coordinate/data processing unit for further processing, wherein in response to a set of indirect texture coordinates the retrieval unit provides retrieved texture data to the processing unit for deriving modified texture coordinates, a method for controlling the texture subsystem to perform one or more indirect texture referencing operations comprising the step of utilizing a generalized indirect-texture referencing API command function to place appropriate instruction codes and/or data in said control register(s), wherein said indirect-texture referencing function may be used to at least:

(i) define up to eight textures stored in a texture memory;

(ii) specify up to eight sets of texture coordinates;

(iii) define up to four indirect texture maps;

(iv) specify up to four indirect texture referencing operations to be performed;

(v) associate one of said eight textures with each indirect texture map; and

(vi) associate one of said eight sets of texture coordinates with each indirect texture maps.

32. In a graphics system having a memory containing texture data, a method of indirect texture referencing comprising the steps of:

(a) using a set of indirect texture-coordinates to retrieve offset data from the memory;

(b) multiplying the offset data by predetermined values forming elements of a texture offset matrix to produce a set of texture offset coordinates; and

(c) using said set of offset texture coordinates for referencing texture data stored in the memory when mapping a predetermined texture to a rendered polygon.

33. A method of indirect texture referencing as in claim 32 wherein said offset data is a result of a predetermined texturing function and comprises at least one texture coordinate offset value.

33. A method of indirect texture referencing as in claim 32 wherein said offset data comprises a set of three values for producing said set of texture offset coordinates.

34. A method of indirect texture referencing as in claim 33 wherein said set of three values comprise s, t and u coordinate offset data.

35. A method of indirect texture referencing as in claim 32 wherein said matrix elements comprise a set of predetermined constants.

31. In a graphics system including a graphics processing pipeline that renders and displays images at least in part in response to polygon vertex data and texture data stored in a memory, the graphics processing pipeline having a texture subsystem for accessing and retrieving texture, the texture subsystem comprising a texture coordinate/data processing unit having: a) at least one binary data multiplier, at least one binary data accumulator and at least one control register for receiving instruction codes and/or data to control texture coordinate/data processing operations, b) a texture data retrieval unit connected to the coordinate/data processing unit, the texture data retrieval unit retrieving texture data stored in a texture memory, and c) a data feedback path from the texture data retrieval unit to the texture coordinate/data processing unit to recycle retrieved texture data through the texture coordinate/data processing unit for further processing, wherein in response to a set of indirect texture coordinates the retrieval unit provides retrieved texture data to the processing unit for deriving modified texture coordinates, a method for controlling the texture subsystem to perform one or more indirect texture referencing operations comprising the step of utilizing a generalized indirect-texture referencing API command function to place appropriate instruction codes and/or data in said control register(s), wherein said indirect-texture referencing function may be used to at least:

(i) define up to eight textures stored in a texture memory;

(ii) specify up to eight sets of texture coordinates;

(iii) define up to four indirect texture maps;

(iv) specify up to four indirect texture referencing operations to be performed;

36. A method of indirect texture referencing as in claim 32 wherein said matrix elements comprise a set of constant and variable values.

37. A method of indirect texture referencing as in claim 32 wherein one or more elements of said matrix are a mathematical function of one or more predetermined texture coordinates.

38. A method of indirect texture referencing as in claim 32 wherein the matrix is a 3 X 2 matrix comprising six predetermined scalar elements.

39. A method of indirect texture referencing as in claim 32 wherein the matrix is arranged having elements as follows:

$$\begin{pmatrix} s/256 & t/256 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined current direct texture coordinates.

40. A method of indirect texture referencing as in claim 32 wherein the matrix is arranged having elements as follows:

$$\begin{pmatrix} 0 & 0 \\ s/256 & t/256 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined current direct texture coordinates.

41. In a graphics system including a graphics engine that renders and displays images at least in part in response to polygon vertex data and texture data stored in an associated memory, a texture processing subsystem for selectively mapping texture data corresponding to one or more different textures and/or texture characteristics to surfaces of said rendered and displayed images, said texture processing subsystem including a texture coordinate offset matrix

arrangement producing a set of offset texture coordinates by multiplying indirect texture data by elements of a matrix, wherein one or more elements of the matrix are a mathematical function of one or more predetermined direct texture coordinates.

42. A texture processing subsystem as in claim 41 wherein the matrix has six elements in three rows and two columns.

43. A texture processing subsystem as in claim 41 wherein the matrix is arranged having elements as follows:

$$\begin{pmatrix} s/256 & t/256 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined current direct texture coordinates.

44. A texture processing subsystem as in claim 41 wherein the matrix is arranged having elements as follows:

$$\begin{pmatrix} 0 & 0 \\ s/256 & t/256 \\ 0 & 0 \end{pmatrix}$$

wherein s and t are predetermined current direct texture coordinates.

45. In a graphics system including a graphics engine that renders and displays images at least in part in response to vertex data and texture data stored in an associated memory, a texture processing subsystem for selectively mapping texture data corresponding to one or more different textures and/or texture characteristics to surfaces of said rendered and displayed images, said texture processing subsystem including a texture coordinate offset matrix arrangement for producing a set of offset texture coordinates by multiplying indirect texture data by

elements of a matrix, wherein one or more elements of the matrix can be selectively loaded.

46. A texture processing subsystem as in claim 45 wherein said elements are predetermined constants.

47. A texture processing subsystem as in claim 45 wherein one or more of said elements are a mathematical function of one or more predetermined direct texture coordinates.

48. A texture processing subsystem as in claim 45 wherein the matrix has six elements in three rows and two columns.

49. In a graphics system having a memory containing texture data, a method of indirect texture referencing comprising the steps of:

(a) using a set of indirect texture coordinates to retrieve texture offset data from memory, said offset data being a result of a predetermined texturing function and comprising at least one texture coordinate offset value;

(b) multiplying said texture offset data by predetermined constant or variable scalar factors; and

(c) using said set of offset texture coordinates for referencing predetermined texture data in memory.

50. A method of indirect texture referencing as in claim 49 wherein said offset data comprise a set of three values for producing said set of texture offset coordinates.

51. A method of indirect texture referencing as in claim 49 wherein said offset data comprise a set of three eight-bit binary values.

52. A method of indirect texture referencing as in claim 49 wherein said offset data comprise a set of three five-bit binary values.

53. A method of indirect texture referencing as in claim 49 wherein said offset data comprise a set of three four-bit binary values.

54. A method of indirect texture referencing as in claim 49 wherein said offset data comprise a set of three three-bit binary values.

55. In a graphics system having a memory containing texture data stored in a texture memory, the texture data accessed using either a set of direct texture-coordinates or a set of indirect texture-coordinates, a method of implementing multiple levels of indirection during indirect texture referencing for mapping a texture onto a primitive, comprising the steps of:

(a) using a set of indirect texture-coordinates to retrieve data triplets stored in texture memory;

(b) deriving a set of modified texture coordinates based at least in part on the retrieved data triplets;

(c) using the set of modified texture coordinates for retrieving data stored in texture memory;

(d) reiteratively repeating steps (b) and (c) for a predetermined number of data retrievals; and

(e) using a set of derived texture coordinates resulting from step (d) to map predetermined texture data onto the primitive.

56. In a graphics system including at least one texture mapping unit, a method of operating the texture mapping unit to provide a plurality of logical texture mapping stages comprising:

(a) presenting a first set of texture mapping parameters to the texture mapping unit;

(b) controlling the texture mapping unit to perform a first texture mapping operation in response to said parameters presented by step (a);

(c) recirculating results of step (b) to develop a further set of texture mapping parameters; and

(d) presenting said further set of parameters to the texture mapping unit and controlling said texture mapping unit to perform a further texture mapping operation in response to said further set of parameters.

57. The method of claim 56 wherein step (b) includes performing an indirect texture mapping operation that develops a data set, recirculating step (c) comprises developing a further set of texture mapping parameters responsive to said developed data set; and step (d) includes performing a direct texture mapping operation in response to the further set of texture mapping parameters.

58. The method of claim 56 wherein the developing step develops said further set of texture mapping parameters by combining a further set of direct texture coordinates with said data set.